

Mecanismos de transporte intracelular: difusión versus transporte dirigido

Raúl Montón¹, dirigido por Fernando Falo²

Resumen

Simulación numérica sobre GPUs para estudiar el movimiento de orgánulos u otros cargos individuales en el citoplasma celular mediante la combinación de procesos de difusión y de transporte activo a través de motores moleculares de kinesinas o dineínas sobre la estructura de microtúbulos, en una configuración radial típica de las células animales no diferenciadas.

¹ Grado en Física, Universidad de Zaragoza

² Departamento de Física de la Materia Condensada, Universidad de Zaragoza

Índice		5	Trabajo futuro	13
	Introducción	1	5.1 Diferentes geometrías	13
1	Métodos	2	5.2 Microtúbulos flexibles	14
1.1	Generación de microtúbulos	2	5.3 Combinación de motores moleculares	14
1.2	Difusión	4	5.4 Modificación del modelo de campo medio	15
1.3	Subdivisión del espacio	4	A Tecnologías utilizadas	16
1.4	Transporte activo	4	B Diseño de la aplicación	16
1.5	Condiciones de contorno	5	C Arquitectura de la aplicación	18
1.6	Colisiones	5	D Algoritmos	19
1.7	Modelo de campo medio	5	E Generación de números aleatorios	20
1.8	Estadística y recolección de datos	6		
1.9	Cálculo en GPUs	6		
2	Análisis de la estructura de microtúbulos	7		
2.1	Distribución de longitudes totales	7		
2.2	Densidad de microtúbulos	7		
3	Resultados del transporte	8		
3.1	Longitud de persistencia variable	9		
3.2	Probabilidad de acoplo variable	9		
3.3	Probabilidad de desacoplo variable . .	10		
3.4	Efectos de crowding	10		
3.5	Comparación con el modelo de campo medio	12		
4	Conclusiones	13		

Introducción

La capacidad de distribuir diferentes tipos de orgánulos y proteínas a determinadas partes de la célula es un requisito indispensable para el buen funcionamiento de la mayoría de células eucariotas. Para ello las células utilizan una combinación de dos mecanismos principales: difusión y transporte por microtúbulos.

Los microtúbulos son estructuras tubulares de pequeño diámetro (≈ 24 nm) [1] [2] formados por proteínas (monómeros de α y β -tubulina). Generalmente se originan en el centrosoma de la célula y se extienden para formar una red que cumple una doble función estructural y de transporte.

Son estructuras con inestabilidad dinámica: mien-

tras uno de los extremos del microtúbulo se mantiene estable (extremo -), en el otro (extremo +) se van añadiendo o retirando monómeros para variar la longitud del microtúbulo. A la desestabilización y rápido desmantelamiento de los monómeros del extremo + se le conoce como *catástrofe*. No obstante, la escala de tiempos en la que esto sucede es lo suficientemente grande como para considerarlos de longitud estática a efectos de transporte. Como referencia, la velocidad de crecimiento de los microtúbulos ha sido determinada experimentalmente para la levadura *Schizosaccharomyces pombe* en ~ 50 nm/s [3] y la de desmantelamiento en de ~ 140 nm/s [3]. En células de mamíferos se ha determinado una velocidad de crecimiento de ~ 66 nm/s [4]. Los motores moleculares son unas proteínas que tienen un sitio de unión a microtúbulo y un sitio de unión a orgánulos o proteínas, desplazándose a lo largo de los primeros. A estos orgánulos o proteínas acoplados al motor molecular los denominaremos genéricamente como *cargos*. Existen varios tipos de motores moleculares, y se desplazan hacia un extremo u otro del microtúbulo en función del tipo que sean. Destacan dos: kinesinas y dineínas. La mayoría de kinesinas transportan sus cargos desplazándose hacia el extremo + del microtúbulo, mientras que la mayoría de dineínas transportan sus cargos hacia el extremo - del microtúbulo. Como mencionábamos antes, la velocidad de los motores moleculares a lo largo de los microtúbulos es considerablemente mayor a la escala de tiempos de la dinámica de microtúbulos, siendo de 814 nm/s [5] para la kinesina-1.

La pérdida de esta capacidad de transporte es la causa de un amplio grupo de enfermedades en organismos pluricelulares, como ciertas enfermedades neurológicas causadas por la desregulación del transporte por microtúbulos a través de los axones [6], e incluso estando bien regulado puede tener un papel relevante en otras enfermedades, como en las causadas por algunos virus que aprovechan la estructura ya existente de microtúbulos para desplazarse por el citoplasma [2].

Así, las estructuras de microtúbulos y los motores moleculares constituyen una parte indispensable

del mecanismo de funcionamiento interno de casi cualquier célula eucariota, en gran parte debido al aumento de tamaño respecto a las células procariotas, que hace de la difusión un mecanismo poco eficiente para el transporte, o el alto nivel de especialización de ciertas líneas celulares en organismos pluricelulares que requieren del transporte por microtúbulos para cumplir su función conjunta. Un ejemplo de esto último sería el de ciertos tejidos endoteliales, donde la disposición apical-basal de microtúbulos tiene una importancia clave en el transporte de un lado a otro de la membrana [7]. Los avances en microscopía de fluorescencia han permitido estudiar cualitativamente [8] las estructuras intactas de microtúbulos en varias líneas celulares, aunque una caracterización pormenorizada de cada microtúbulo aún se escapa a las capacidades actuales de este método. Así, varios modelos de generación procedural de microtúbulos han sido propuestos [9] para intentar reproducir las densidades de microtúbulos obtenidas en las imágenes de microscopía de fluorescencia y estudiar así su estructura.

En este trabajo partimos de uno de estos métodos procedurales de generación de microtúbulos, que nos proporciona una estructura de microtúbulos realista, y lo combinamos con una simulación de la dinámica de los orgánulos que son transportados por difusión simple o a través de microtúbulos, bajo unas condiciones de contorno periódicas que facilitan la comparación del sistema con posibles modelos analíticos más fácilmente computables.

1. Métodos

1.1 Generación de microtúbulos

Para la generación de los microtúbulos usamos una variación del método propuesto en [9]. Para generar cada microtúbulo partimos de un punto inicial aleatorio generado con distribución uniforme en el interior de una esfera de pequeño radio (equivalente al centrosoma de una célula) que elegimos con $R_{\text{centrosoma}} = 1200$ nm. El segundo punto del microtúbulo se genera en dirección radial (opuesta al centro del centrosoma).

A partir de este segundo punto, los siguientes se

generan sobre la superficie de un cono de ángulo α cuyo eje principal apunta en la dirección del segmento anterior. El ángulo de rotación (comúnmente denominado ϕ) es aleatorio, y el nuevo punto generado se sitúa siempre a una misma distancia del anterior, a la que llamaremos d_{MT} o longitud del segmento de microtúbulo. Su valor está fijado en 50 nm.

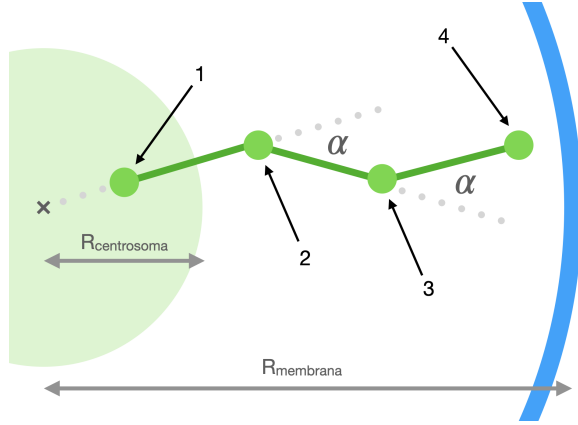


Figura 1. Representación del algoritmo de generación de microtúbulos. **1.** El primer punto se genera uniformemente dentro del centrosoma. **2.** El segundo punto se genera en dirección radial. **3.** Se generan los siguientes puntos en la superficie del cono de ángulo α alineado con el segmento anterior. **4.** No se genera ningún punto después de este porque quedaría fuera de la membrana.

Los microtúbulos se consideran completos (y por tanto dejan de añadirse puntos) cuando llegan lo suficientemente cerca a la membrana celular, generada como una esfera de $R_{membrana} = 12000$ nm, como para que el siguiente segmento de microtúbulo a generar quedase fuera de ella.

A la hora de comparar los parámetros del modelo de generación de microtúbulos podemos usar la colinealidad, definida como $\cos(\alpha)$, o la longitud de persistencia, que se puede calcular como [10] [11]:

$$\langle \cos(\theta) \rangle = e^{-L/\ell_P} \quad (1)$$

Donde θ es el ángulo entre el primer vector del microtúbulo y otro punto del mismo microtúbulo a una distancia L recorrida a lo largo del mismo y ℓ_P es la longitud de persistencia. Dado que el

cálculo involucra el valor esperado de $\cos(\theta)$ es necesario calcularla en puntos a varias distancias y hacer un fit a una recta utilizando la pendiente del decaimiento inicial de $\langle \cos(\theta) \rangle$ como se detalla en [11].

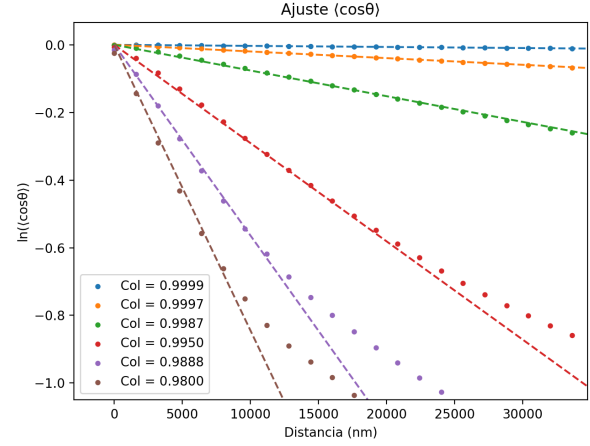


Figura 2. Ajuste de $\ln(\langle \cos(\theta) \rangle)$ para obtener la longitud de persistencia ℓ_P .

Esta magnitud presenta ventajas sobre la colinealidad a la hora de establecer comparaciones, pues es independiente del tamaño del segmento elegido para el modelo de generación de microtúbulos y es un parámetro que ya se ha obtenido experimentalmente, variando entre $\approx 20 \mu m$ *in vivo* [12] y $\approx 300 \mu m$ *in vitro* [13], proporcionándonos un marco de posibles valores con significado biológico.

Con esto podemos construir una tabla que relacione los valores del ángulo local α , su respectiva colinealidad $\cos(\alpha)$ y la longitud de persistencia ℓ_P determinada, para el modelo con longitud de segmento $d_{MT} = 50$ nm.

α	$\cos(\alpha)$	ℓ_P (μm)
0.000	1.0000	∞
0.010	0.9999	3135.2
0.025	0.9997	507.2
0.050	0.9987	131.6
0.100	0.9950	34.4
0.150	0.9888	17.7
0.200	0.9800	11.8

Otro parámetro interesante a calcular es la densidad de microtúbulos. Dado que nuestro modelo de microtúbulos está discretizado en una serie de puntos,

consideraremos que cada uno de ellos equivale a un segmento de microtúbulo de longitud $d_{MT} = 50$ nm a efectos de cómputo de densidades, utilizando la densidad lineal de los microtúbulos, ≈ 160000 Da/nm tabulada en [14].

1.2 Difusión

Los orgánulos se transportan por difusión simple cuando no están siendo transportados a través de los microtúbulos. Durante una delta de tiempo δt la distancia media recorrida por difusión viene dada por la ley de Fick [15]:

$$\langle \delta r \rangle = \sqrt{6D \delta t} \quad (2)$$

Donde D es el coeficiente de difusión de una esfera 150 nm de radio, correspondiente al tamaño de los orgánulos, en un medio con viscosidad η de acuerdo con la relación de Stokes-Einstein:

$$D = \frac{k_B T}{6\pi\eta r} \quad (3)$$

Donde r es el radio del orgánulo, η la viscosidad del medio y T la temperatura, fijada a 300K. Para tener en cuenta el efecto del *crowding* (ver Apartado 3.4) por la presencia de otros orgánulos o moléculas en el citoplasma no utilizaremos la viscosidad típica del agua pura (η_{agua}) sino que introducimos el parámetro η_w que modula la viscosidad del medio como un múltiplo de la viscosidad del agua (es decir, $\eta_w = 1$ para agua pura sin efectos de *crowding*). Así, la ecuación 2 pasa a estar definida como:

$$D = \frac{k_B T}{6\pi\eta_{agua}\eta_w r} = \frac{1,59349 \cdot 10^6}{\eta_w} \quad (4)$$

Donde $1,59349 \cdot 10^6$ es el coeficiente de difusión del orgánulo en agua pura, ya precomputado, y modificable fácilmente mediante el parámetro η_w .

En la simulación modelamos la difusión como una caminata aleatoria que se mueve una distancia promedio $\langle \delta r \rangle$ en cada paso temporal, según:

$$r = \sqrt{6D \delta t} \cdot \left(\frac{u_x, u_y, u_z}{\sqrt{0,5^2 + 0,5^2 + 0,5^2}} \right) \quad (5)$$

Donde las u_i son variables aleatorias independientes extraídas de una distribución uniforme en $[0, 1)$ y el denominador ajusta la expresión para tener norma promedio igual a 1.

Para seleccionar δt , el paso temporal utilizado en la simulación, propondremos que los orgánulos se muevan en promedio una distancia comparable con su propio tamaño (150 nm), fijando así $\langle \delta r \rangle \approx 150nm$, que a su vez fija el valor de δt según la ecuación 2. La distancia recorrida en los microtúbulos en un paso temporal será menor que la $\langle \delta r \rangle$ de un paso de difusión salvo para valores muy altos del coeficiente de viscosidad del medio, η_w , siendo así difusión la que limita la máxima δt aceptable.

1.3 Subdivisión del espacio

A fin de facilitar los cálculos a realizar durante la evolución temporal de la simulación inscribimos la esfera que contiene a la célula en un cubo y subdividimos el espacio en celdas cúbicas de tamaño uniforme, a razón de 100 celdas por lado. Cada una de estas celdas tendrá un identificador único.

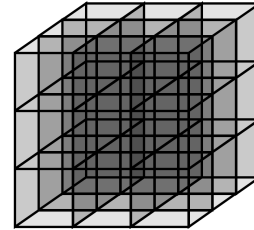


Figura 3. Subdivisión del espacio en células cúbicas idénticas donde está precomputado el número de puntos de acoplo a microtúbulos.

1.4 Transporte activo

En la simulación los microtúbulos están representados como una serie de puntos. Después de un tiempo $N\delta t$, es decir, cada N pasos temporales de difusión, las partículas son transportadas de un punto del microtúbulo al siguiente, de forma que la velocidad $v_{MT} = d_{MT} / N\delta t$ sea la velocidad experimental estimada para el transporte en microtúbulos, donde d_{MT} es la longitud de un segmento de microtúbulo.

La probabilidad de que una partícula difundiendo se acople a un microtúbulo o de que una partícula siendo transportada activamente pase a ser difundida es proporcional a unos parámetros w_{ON} y w_{OFF} . En particular, la probabilidad de que un orgánulo transportado a través de un microtúbulo pase a difundirse es:

$$P_{detach} = w_{off} \delta t \quad (6)$$

Dicho parámetro está fijado en $w_{off} = 1 s^{-1}$ en todo momento, un valor ligeramente inferior al determinado experimentalmente para la kinesina-1 ($1,38 \pm 0,29 s^{-1}$) [16] y la kinesina-2 ($2,00 \pm 0,44 s^{-1}$) [16] en disoluciones saturadas de ADP.

Por otro lado, la probabilidad de que se acople a un microtúbulo es:

$$P_{attach} = 1 - \left(1 - \frac{w_{on} \delta t}{V_{celda}}\right)^N \quad (7)$$

Donde V_{celda} es el volumen de las celdas en las que está subdividido el espacio y N es el número de puntos de microtúbulo presentes en esa celda. La expresión completa es la probabilidad (binomial) de que el orgánulo se acople a al menos uno de los puntos de unión al microtúbulo de la celda en la que se encuentra en un paso de simulación (de duración δt).

Cuando el orgánulo llega al último punto de un microtúbulo pasa automáticamente a transportarse por difusión. Además, se impide que los orgánulos puedan acoplarse al último punto de un microtúbulo para evitar que entren en un bucle de desacoplo-acoplo inmediato al final de los microtúbulos en simulaciones con w_{on} alto.

1.5 Condiciones de contorno

Para facilitar la comparación con modelos analíticos, la simulación utiliza unas condiciones de contorno periódicas. En el caso de las kinesinas, que mueven las partículas hacia el exterior de la célula, cuando estas exceden un determinado radio, correspondiente a la posición de la membrana celular, son reinyectadas automáticamente y con distribución uniforme en el centrosoma, una esfera de menor radio generalmente situada en el centro de la célula.

En el caso de las dineínas las condiciones de contorno son las contrarias: las partículas que se adentran en el centrosoma son reinyectadas en la superficie exterior de la célula, correspondiente a la membrana celular.

1.6 Colisiones

Para evitar tener que serializar completamente la dinámica de las partículas (que impediría hacer los cálculos en la GPU de forma efectiva) se calculan en paralelo las nuevas posiciones tentativas de todas las partículas tras un paso temporal realizando sus respectivos movimientos de difusión o transporte por microtúbulos como si fuesen totalmente independientes entre sí, y la celda del espacio correspondiente a esa nueva posición.

Una vez calculadas las nuevas posiciones y celdas una segunda función itera sobre todas las partículas de cada célula (las células siguen recorriéndose en paralelo, pues sus partículas son independientes de las de otras células) y mueve secuencialmente las partículas de su posición en el paso temporal anterior a la posición precalculada para el paso temporal siguiente como si fueran independientes. Si alguna de estas nuevas posiciones se encontrase en una celda que ya estuviera ocupada, ya sea por una partícula que se ha movido a su nueva posición o por una que aún no ha sido alcanzada por el bucle de actualización, dicha partícula mantendrá su posición actual, que naturalmente será una celda que no puede haber sido ocupada (salvo por sí misma), considerando así el efecto del volumen excluido de las partículas sin tener que serializar completamente la simulación.

Durante la reinyección se ignoran las colisiones, ya que de otro modo podrían darse situaciones en las que fuese imposible mantener las condiciones de contorno. No obstante, una vez reinyectados, los orgánulos deben seguir de nuevo las restricciones impuestas por las colisiones, de modo que solo se moverán a celdas libres.

1.7 Modelo de campo medio

Esta simulación puede intentar aproximarse por un modelo de campo medio, gobernado por las

ecuaciones [15]:

$$\frac{\partial C_d}{\partial t} = D \nabla^2 C_d - k_{att} C_d [m] + k_{det} C_m \quad (8)$$

$$\frac{\partial C_m}{\partial t} = -\nabla \cdot (v C_m) + k_{att} C_d [m] - k_{det} C_m \quad (9)$$

Donde C_d es la concentración de partículas en difusión, C_m es la concentración de partículas siendo transportadas por los microtúbulos y m la concentración de microtúbulos, que aproximaremos por una distribución proporcional a $1/r^2$ (microtúbulos totalmente rectos). k_{att} y k_{det} son las probabilidades de acoplo y desacoplo de los microtúbulos y v es la velocidad de transporte a través de los microtúbulos. Las condiciones de contorno serán análogas a las fijadas para la simulación: los orgánulos transportados fuera de la célula (por difusión o transporte dirigido) son reinyectadas inmediatamente en el centrosoma. Así:

$$C_d(R_2) = 0 \quad (10)$$

$$R_1^2 \nabla C_d(R_1) = R_2^2 \nabla C_d(R_2) \quad (11)$$

$$R_1^2 C_m(R_1) = R_2^2 C_m(R_2) \quad (12)$$

Donde R_1 y R_2 son el radio del centrosoma y el radio de la membrana celular. La solución a estas ecuaciones se obtiene numéricamente utilizando el algoritmo Runge-Kutta de segundo orden [17]. La discretización temporal y espacial del sistema se realiza cumpliendo el criterio de Courant:

$$v \frac{\Delta t}{\Delta r} < 1 \quad D \frac{\Delta t}{(\Delta r)^2} < 1/2 \quad (13)$$

1.8 Estadística y recolección de datos

Para obtener unos resultados que no se vean alterados por anisotropías locales en la disposición de los microtúbulos generados para una célula concreta, se crean varias células separadas con idénticos parámetros de generación de microtúbulos (pero no idéntica estructura de microtúbulos para cada célula) y se ejecuta la simulación en paralelo en

todas ellas. Así, las figuras obtenidas no representan solo un promedio de varias partículas en una misma célula, sino que están promediadas también entre varias células con los mismos parámetros.

Además, dado que la distribución de posiciones de orgánulos adquiere con el tiempo una distribución estacionaria, y con el fin de reducir aún más el ruido estadístico en los datos obtenidos, se realiza también un promedio temporal de dicha distribución una vez ha alcanzado su estado de equilibrio.

Salvo que se mencione explícitamente un cambio en estos parámetros, las figuras obtenidas son la simulación de $4 \cdot 10^5$ orgánulos repartidos entre 200 células (2000 partículas/célula). Las simulaciones se estabilizan durante 240 s antes de recopilar los datos.

1.9 Cálculo en GPUs

Para poder manejar estos cálculos en una cantidad razonable de tiempo recurrimos al cálculo en GPUs. La simulación propuesta es fácilmente paralelizable, especialmente en el caso en el que no se tienen en cuenta las colisiones entre orgánulos, haciéndola idónea para ser calculada en la GPU. Para ello utilizaremos la API Metal que permite hacer cálculos en la GPU en dispositivos con iOS y macOS.

Resulta difícil calcular con exactitud la diferencia que supone realizar el cálculo en la GPU frente a la CPU sin reescribir completamente la simulación. Nos limitamos a realizar una estimación con un modelo en Python que realiza, para cada partícula, las operaciones más costosas de la simulación real (generación de números aleatorios, cálculo de distancias e indexado de la matriz que contiene los datos de microtúbulos para cada celda). Así, obtenemos una cota inferior del tiempo que se tardaría en ejecutar la simulación. Para $4 \cdot 10^5$ partículas, el número de pasos de simulación que se pueden realizar en un segundo es de ≈ 4.5 pasos/segundo en la CPU y de ≈ 80 pasos/segundo en la GPU, siendo por tanto esta última en torno a 20 veces más rápida como mínimo.

2. Análisis de la estructura de microtúbulos

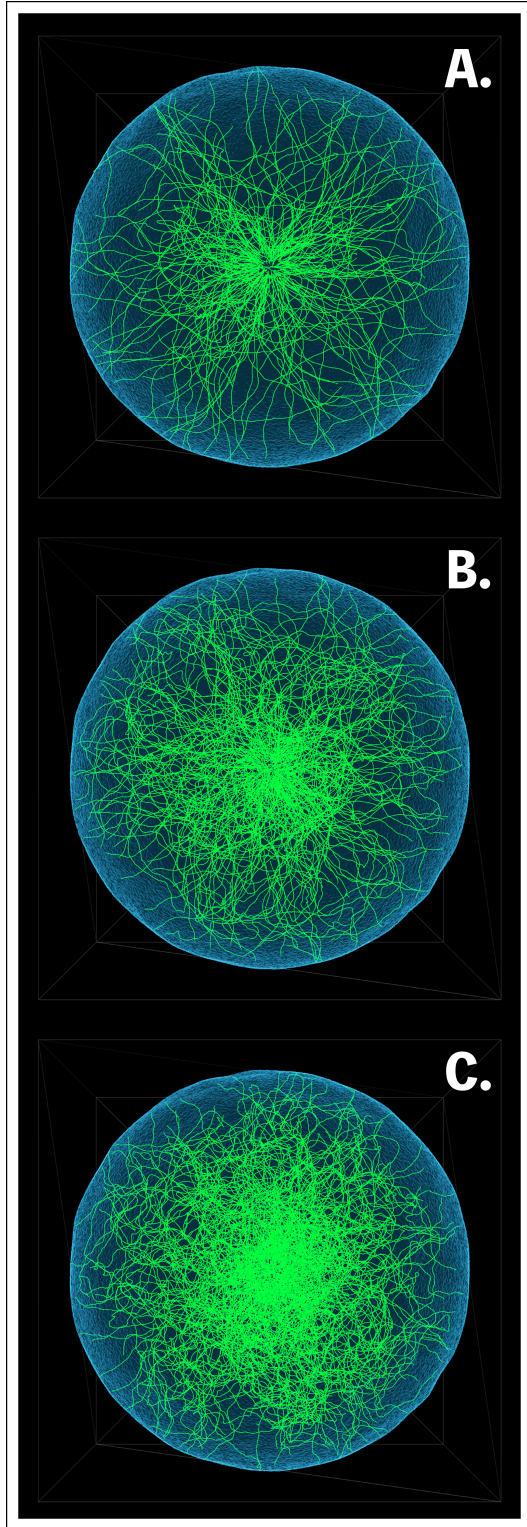


Figura 4. Estructura de microtúbulos generada para tres longitudes de persistencia **A.** $\ell_P = 34.4 \mu\text{m}$, **B.** $\ell_P = 17.7 \mu\text{m}$, **C.** $\ell_P = 11.8 \mu\text{m}$.

Por su similitud con los valores obtenidos *in vivo*, analizamos los resultados para tres longitudes de persistencia distintas: 34.4, 17.7 y 11.8 μm , que podemos ver en la Figura 4. En ella se puede apreciar cualitativamente el aumento de la densidad en la zona central de la célula y la rigidez aparente de los microtúbulos, que aparecen más curvados cuanto más pequeña es la longitud de persistencia.

2.1 Distribución de longitudes totales

Midiendo las distribuciones de longitud de los microtúbulos generados con el algoritmo mencionado en el apartado 1.1 obtenemos los resultados de la Figura 5.

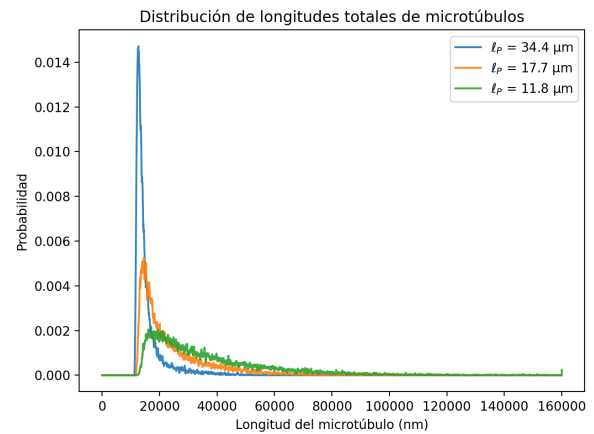


Figura 5. Distribución de probabilidad de longitud total de microtúbulo para varias longitudes de persistencia.

Como cabría esperar, para longitudes de persistencia muy altas (microtúbulos prácticamente rectos) observamos que la mayoría de microtúbulos tienen una longitud total en torno al valor del radio de la célula (12000 nm). A medida que generamos microtúbulos con menor longitud de persistencia el valor medio de la longitud total de los microtúbulos se desplaza hacia valores más grandes, al tiempo que aumenta su desviación estándar.

2.2 Densidad de microtúbulos

Las estructuras de microtúbulos generadas tendrán una densidad proporcional a $1/r^2$ para $\alpha = 0$ (longitud de persistencia infinita), desviándose de esta expresión a medida que disminuye la longitud de

persistencia. No obstante, las densidades generadas siguen pudiendo ajustarse a una distribución proporcional a $1/r^v$. Los valores ajustados de v utilizando los microtúbulos generados en la simulación podrían utilizarse para modificar las ecuaciones de campo medio y mejorar su estimación de la densidad de microtúbulos en una célula real.

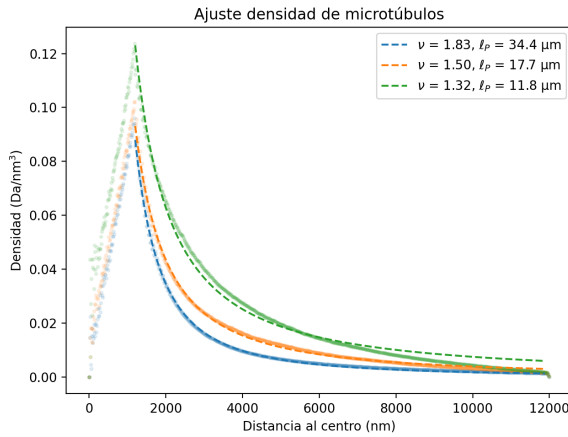


Figura 6. Ajuste de la densidad de microtúbulos generados por el modelo para varias longitudes de persistencia.

Observamos que para las dos longitudes de persistencia más altas el ajuste a una curva de la forma $1/r^v$ es válido, mientras que para $\ell_P = 11,8 \mu\text{m}$ se subestima notablemente la densidad en la zona central de la célula al tiempo que se sobreestima en el borde exterior.

En cualquier caso, vemos que incluso en las zonas más densas la densidad no supera los $0,12 \text{ Da/nm}^3$, de modo que el modelo no excede la máxima densidad físicamente posible (lo que indicaría que se están creando demasiados microtúbulos).

3. Resultados del transporte

Como resultados del transporte vamos a calcular, modificando cada vez uno de los parámetros de la simulación, la distribución de posiciones de los orgánulos en el citoplasma celular y la distribución de tiempos de llegada del centrosoma a la membrana celular. La primera es un indicador de la capacidad de la célula de redirigir los orgánulos a una zona particular de la misma, mientras que la

segunda es un indicador de la eficiencia del transporte.

En la Figura 7 podemos observar una representación directamente extraída del programa donde vemos, para dos células generadas con los mismos parámetros de generación de microtúbulos, la distinta distribución de los orgánulos. Para acentuar la diferencia utilizamos un w_{on} bajo para la primera figura (de modo que la mayoría de orgánulos se encuentran difundiendo) y un w_{on} muy alto para la segunda figura, de modo que la mayoría de orgánulos estén siendo transportados por motores moleculares.

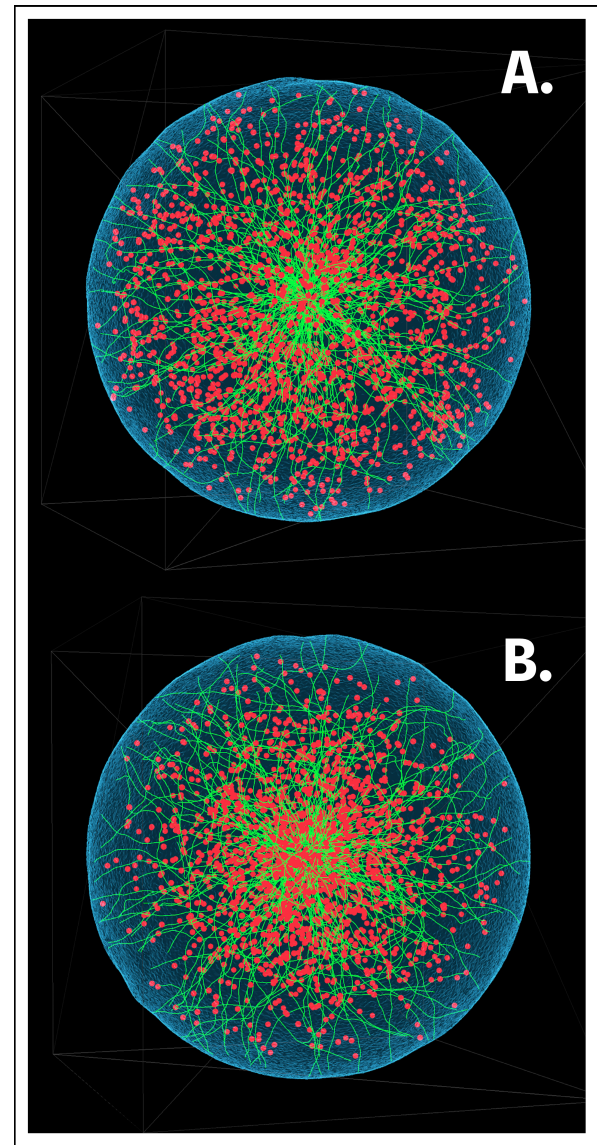


Figura 7. Visualización de las células generadas en la simulación incluyendo la estructura de microtúbulos y 2000 orgánulos por célula.

3.1 Longitud de persistencia variable

3.1.1 Distribución de posiciones de orgánulos

Con las estructuras de microtúbulos generadas observamos cómo varía la distribución de las partículas en el interior de la célula (Figura 8), donde la línea superior de cada color representa el total de orgánulos para una determinada longitud de persistencia y la línea inferior el número de orgánulos que están siendo transportados en microtúbulos.

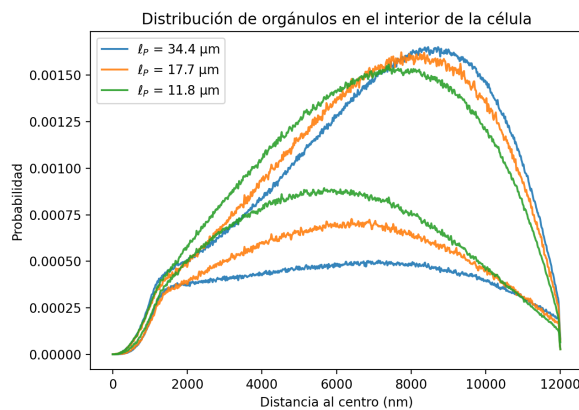


Figura 8. Distribución de probabilidad de distancia al centro de la célula para un orgánulo cualquiera utilizando varias longitudes de persistencia diferentes.

Se observa que a medida que disminuye la longitud de persistencia aumenta el número de orgánulos que son transportados activamente. Esto es así ya que aumenta considerablemente la densidad de microtúbulos (y con ello el número de puntos de unión para que los orgánulos se acoplen al microtúbulo), especialmente en la zona más central de la célula, dado que en la zona más externa los microtúbulos que crucen el umbral de la membrana celular dejan de crecer automáticamente.

La causa de este comportamiento la podemos ver en la Figura 6, donde representamos la densidad de microtúbulos en función del radio para varias longitudes de persistencia distintas. En particular, observamos que la densidad aumenta en mayor medida en la zona entre ≈ 2000 y ≈ 5000 nm, explicando la variación en las distribuciones de posiciones de los orgánulos obtenidas en la Figura 8.

3.1.2 Distribución de tiempos de llegada

Vemos también cómo varían las distribuciones de tiempos de llegada con la longitud de persistencia. Así, obtenemos los resultados de la Figura 9.

El tiempo medio de llegada aumenta al disminuir la longitud de persistencia. En general, la mayoría de los tiempos de llegada (aunque no todos) son superiores a la cota del tiempo que le costaría a una partícula recorrer la distancia centrosoma-membrana a través de un microtúbulo perfectamente recto de forma ininterrumpida.

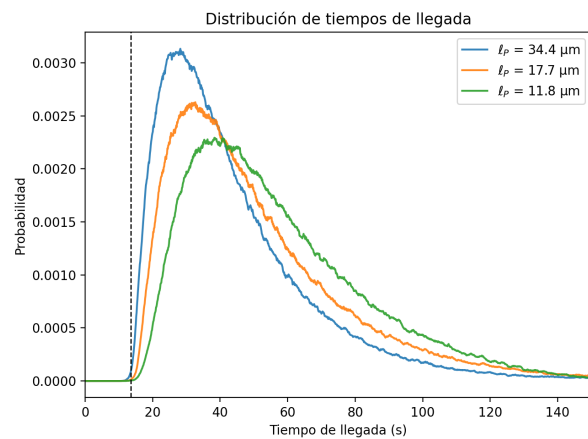


Figura 9. Distribución de tiempos de llegada para varias longitudes de persistencia.

3.2 Probabilidad de acoplo variable

Otro de los parámetros variables de la simulación es el w_{on} , que modula la probabilidad de que un orgánulo dado se acople a un punto de anclaje del microtúbulo.

3.2.1 Distribución de posiciones de orgánulos

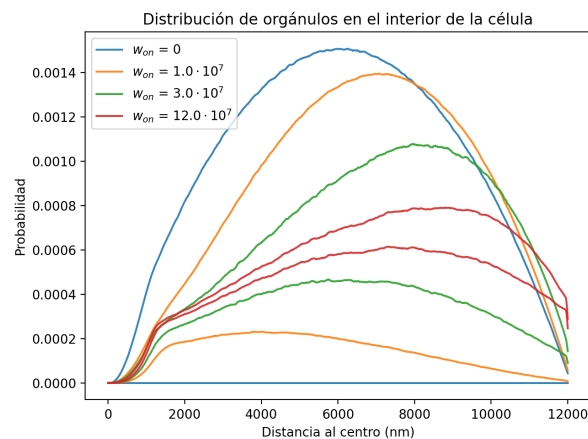


Figura 10. Distribución de posiciones de orgánulos para varios w_{on} distintos.

Vemos que el aumento del parámetro w_{on} provoca que el máximo de la distribución se desplace progresivamente hacia la derecha, al tiempo que se aplana la curva. Esto entra dentro de lo esperado, pues ambos límites del valor de w_{on} , 0 e ∞ , tienen solución analítica conocida. Para $w_{on} = 0$ la distribución es la de difusión pura y la solución analítica coincide con la obtenida. Para $w_{on} \rightarrow \infty$ y $\ell_P \rightarrow \infty$ la distribución debería ser totalmente plana, pues los orgánulos se distribuirían con exactamente la misma distribución que los microtúbulos.

3.2.2 Distribución de tiempos de llegada

Para los tiempos de llegada, observamos que a medida que aumenta w_{on} disminuye el tiempo medio de llegada. Así, el transporte es más eficiente cuanto más alta es la probabilidad de acoplo.

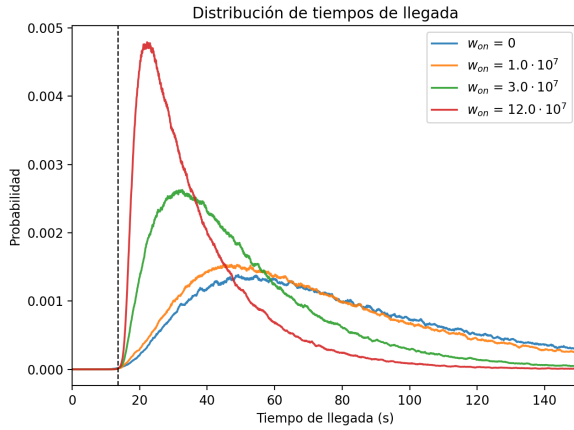


Figura 11. Distribución de tiempos de llegada para varios w_{on} .

3.3 Probabilidad de desacoplo variable

Otro de los parámetros variables de la simulación es el w_{off} , que modula la probabilidad de que un orgánulo siendo transportado por un motor molecular a lo largo de un microtúbulo se desacople del mismo y pase a difundirse. En el resto de simulaciones hemos fijado este parámetro como $w_{off} = 1 \text{ s}^{-1}$ (ver Apartado 1.4), aunque ahora vamos a calcularlo también para $w_{off} = 0.1 \text{ s}^{-1}$ y $w_{off} = 10.0 \text{ s}^{-1}$.

3.3.1 Distribución de posiciones de orgánulos

Vemos en la Figura 12 que para w_{off} bajos la inmensa mayoría de orgánulos están siendo transportados por los microtúbulos. A valores bajos de w_{off}

la distribución de posiciones se acerca a la de difusión pura, aunque se mantiene un pequeño número de partículas siendo transportadas activamente en la zona más interna de la célula.

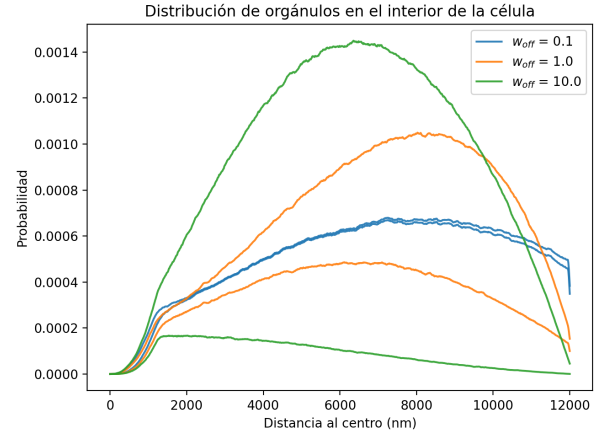


Figura 12. Distribución de posiciones de orgánulos para varios w_{off} distintos.

3.3.2 Distribución de tiempos de llegada

En cuanto a los tiempos de llegada, vemos que un mayor w_{off} aumenta considerablemente los tiempos medios de llegada.

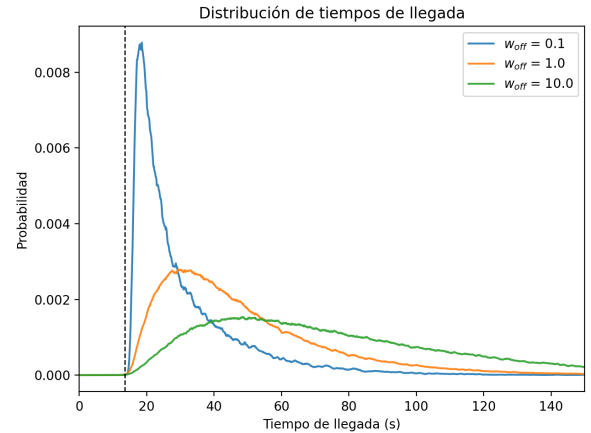


Figura 13. Distribución de tiempos de llegada para varios w_{off} .

3.4 Efectos de crowding

El abarrotamiento o *crowding* es la modificación que se produce en el transporte debido a la presencia de otros orgánulos, proteínas u otras moléculas en el citoplasma. La consecuencia de este abarrotamiento es un aumento efectivo de la viscosidad del medio en el que se mueven los orgánulos, que introducimos en nuestras simulaciones por la modificación del coeficiente de difusión mediante el

parámetro η_w . Por otra parte tenemos que incluir también el efecto de volumen excluido entre los orgánulos que se transportan. Este efecto va a ser especialmente relevante en el transporte a través de microtúbulos ya que puede inducir a la formación de atascos.

La viscosidad del medio controla la velocidad de difusión de los orgánulos y condensa el efecto de *crowding* de todos los orgánulos, proteínas y otras macromoléculas en el citoplasma, que por efecto de su volumen aumentan la viscosidad del medio. Otra modelización más explícita de este fenómeno es el efecto del volumen excluido entre los propios orgánulos simulados, añadiendo colisiones entre las partículas de forma que no puedan ocupar un mismo espacio. Esto modifica también la velocidad efectiva de difusión, pues si hay un choque entre dos orgánulos uno de ellos vuelve a su posición previa al choque, ralentizando la difusión. No obstante, el efecto esperado aquí es menor dado que el volumen combinado de todos los orgánulos es mucho menor al volumen total de la célula ($< 1\%$), y las colisiones de partículas en difusión deberían ser relativamente raras. El motivo por el que se añade es su posible efecto sobre el transporte por microtúbulos, dado que el espacio disponible para acoplarse o desplazarse por los microtúbulos pasa a estar limitado por el de otras partículas que ya estén acopladas, dando lugar a una especie de atascos o embotellamientos en los microtúbulos no contemplados variando únicamente la viscosidad. Así, en la simulación hemos utilizado típicamente un valor $\eta_w = 10$ para tener en cuenta los efectos de *crowding*, un valor intermedio entre los obtenidos en [18]. No obstante, vamos a calcular los efectos de modificar este parámetro en un rango que cubra la mayoría de valores razonables. En particular, vemos los resultados para $\eta_w = 1$ (agua pura), $\eta_w = 10$ (el valor utilizado en el resto de resultados) y $\eta_w = 100$.

Para observar mejor el efecto de las colisiones, en sus respectivas gráficas (Figuras 15 y 17) utilizamos un sistema con relativamente pocos microtúbulos ($N = 50$) y un parámetro w_{on} alto ($6 \cdot 10^7$). Así, intentamos propiciar que se produzca un atasco de

los microtúbulos cuyo efecto sea apreciable en las gráficas de distribución de posiciones de orgánulos y tiempos de llegada. Para $N = 200$ (el número de microtúbulos utilizado en el resto de figuras) las gráficas son tan similares que la diferencia es inferior al ruido.

3.4.1 Distribución de posiciones de orgánulos

En cuanto al efecto sobre las distribuciones de posiciones de los orgánulos, el efecto del aumento de la viscosidad (Figura 14) es mucho más notorio que el causado por activación de las colisiones (Figura 15), prácticamente inapreciable pese a haber utilizado unos parámetros que deberían potenciar este último.

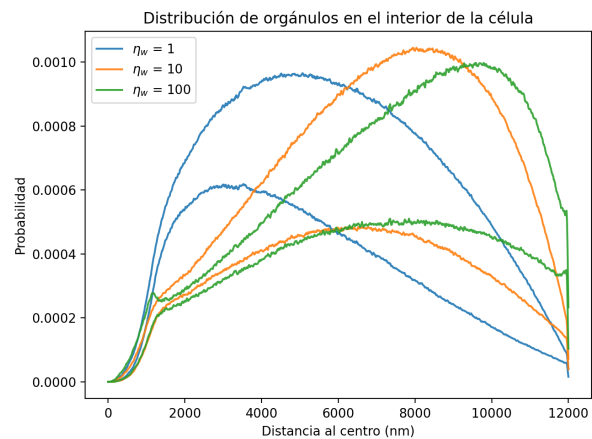


Figura 14. Distribución de posiciones de los orgánulos en el interior de la célula para tres viscosidades del medio diferentes.

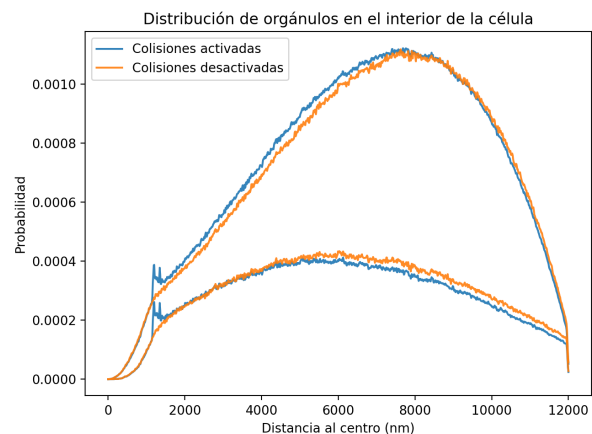


Figura 15. Distribución de posiciones de los orgánulos en el interior de la célula con y sin las colisiones activadas.

En la distribución de $\eta_w = 100$ observamos dos pe-

queños picos en los extremos de los microtúbulos. Estos son causados debido a que la difusión es tan lenta que los microtúbulos tardan mucho tiempo en acoplarse al inicio de los microtúbulos (el pico de la izquierda) y en difundirse hasta la membrana celular desde que se sueltan del final del microtúbulo. En la distribución con colisiones (Figura 15) encontramos un pico similar, pero en este caso está causado por el embotellamiento al inicio de los microtúbulos, donde la densidad de los mismos es más elevada.

3.4.2 Distribución de tiempos de llegada

En cuanto a los efectos en la distribución de tiempos de llegada (Figura 16), nuevamente el efecto de la viscosidad es notoriamente más grande que el efecto del volumen excluido (Figura 17) .

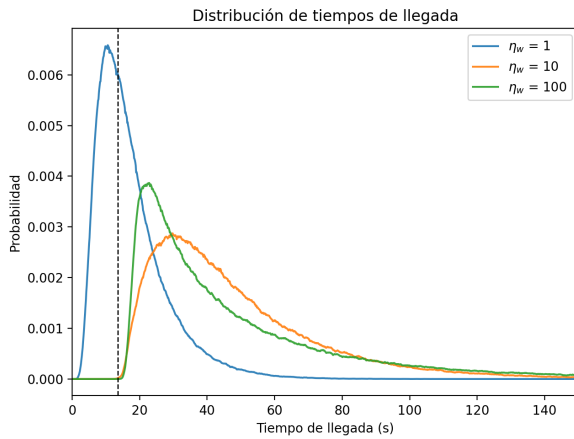


Figura 16. Distribución de tiempos de llegada para dos viscosidades del medio diferentes.

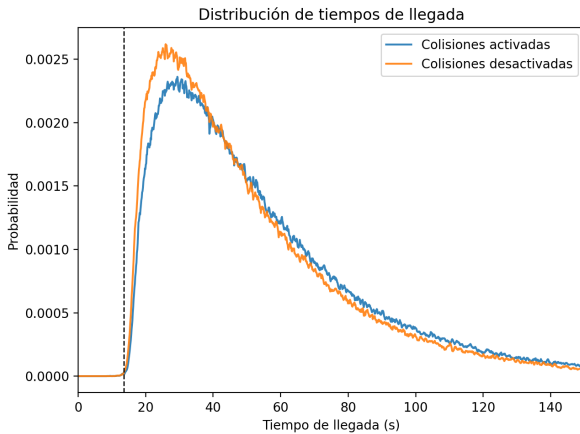


Figura 17. Distribución de tiempos de llegada con y sin las colisiones activadas.

En el caso de los tiempos de llegada para $\eta_w = 1$ vemos que buena parte de los tiempos de llegada son más cortos que el tiempo mínimo de llegada por microtúbulos, algo que no ocurre para ninguna combinación de los parámetros ℓ_P , w_{on} o w_{off} para $\eta_w = 10$.

3.5 Comparación con el modelo de campo medio

Comparando los resultados del modelo de campo medio con la simulación obtenemos los resultados de Figura 18, de donde se puede deducir que para los valores del número de microtúbulos (200) que hemos empleado en la simulación el modelo de campo medio no presenta un buen ajuste. La explicación para esta diferencia parece venir de que el modelo de campo medio asume una densidad de microtúbulos constante para un radio dado (es decir, para cada una de las cortezas esféricas en las que se discretizan las ecuaciones) mientras que en la simulación la probabilidad de acoplo a microtúbulos depende de la densidad local de puntos de unión a microtúbulos y por tanto no es constante dentro de una misma corteza esférica.

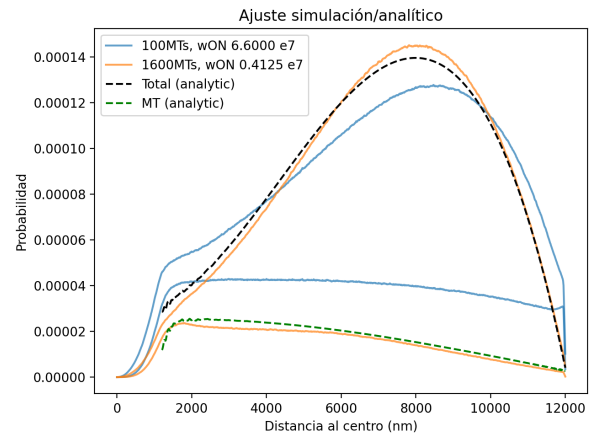


Figura 18. Ajuste de la simulación al modelo de campo medio para dos valores extremos de microtúbulos.

Para validar esta hipótesis, aumentamos el número de microtúbulos bajando proporcionalmente el w_{on} para que la probabilidad de acoplo se mantenga constante. Al aumentar el número de puntos de unión a microtúbulo su densidad será más ho-

mogénea dentro de una misma corteza esférica (y la probabilidad se mantendrá igual por la disminución del w_{on}) y los resultados deberían aproximarse más al modelo de campo medio. Además, para descartar que se trate de un efecto de la menor colinealidad de los microtúbulos generados frente a la suposición de microtúbulos perfectamente colineales con $\rho \propto 1/r^2$ del modelo de campo medio, se ha utilizado una longitud de persistencia infinita, de modo que la densidad de microtúbulos en la simulación se ajuste perfectamente con el modelo analítico para que las condiciones sean lo más cercanas posibles al modelo de campo medio.

4. Conclusiones

La aplicación desarrollada permite estudiar la dinámica de los orgánulos en el interior de la célula sin necesidad de modificaciones en el código para la variación de los parámetros. La herramienta de visualización resulta muy útil a la hora de interpretar los resultados numéricos de la simulación, combinando la representación en tiempo real de los orgánulos y microtúbulos de una de las células simuladas con los gráficos, también en tiempo real, del promedio de todas las células simuladas. Esto permite inspeccionar visualmente algunos efectos como el *crowding* y entender mejor cómo está afectando a los resultados, al tiempo que facilita la validación de los cambios que se introduzcan en el programa.

Además, la dinámica de los orgánulos se calcula en paralelo utilizando todos los núcleos de la GPU, lo que aumenta en varios órdenes de magnitud la cantidad de datos recopilados, permitiendo comparar algunos resultados que habrían quedado ocultos por las fluctuaciones estadísticas (como el efecto de las colisiones) realizando un cálculo tradicional en CPUs, mucho más lento.

Esta aplicación desarrollada es de código abierto y está disponible en <https://github.com/Androp0v/CellTransport>.

Comprobamos también que los cambios en la longitud de persistencia de los microtúbulos redistribuyen los orgánulos hacia una zona u otra de la célula.

Así, para microtúbulos más rígidos hay una (relativamente) mayor presencia de orgánulos cerca del borde de la célula mientras que para microtúbulos menos rígidos los orgánulos aumentan su presencia en la zona central de la célula.

Los parámetros w_{on} y w_{off} también juegan un papel en la distribución de los orgánulos en el interior de la célula. Aunque en este trabajo no se han simulado los motores moleculares individualmente, su número podría ser uno de los factores que modificase el parámetro efectivo w_{on} o w_{off} al tiempo que serían fácilmente regulables por la célula variando la producción de los mismos.

Además, apreciamos diferencias entre la simulación y el modelo analítico, encontrando variación significativa en el ajuste con la modificación de alguno de los parámetros. La validez del modelo analítico se asienta en un régimen donde el número de microtúbulos es relativamente elevado. Los ejemplos biológicos donde el número de microtúbulos se ha determinado experimentalmente parecen tener menos de los necesarios para poder suplir completamente la necesidad de simulaciones de N-partículas con modelos de campo medios. Por ejemplo, en las células HeLa (una línea de células tumorales humanas) el número de microtúbulos varía entre 25 y 300 [9], mientras que en los *Plasmodium* (un tipo de parásito eucariota unicelular) alcanza un máximo de ~ 16 [19], ambos ejemplos muy lejos del umbral de los 1600 microtúbulos.

5. Trabajo futuro

Tras haber validado el correcto funcionamiento del modelo básico el modelo podría extenderse para cubrir ejemplos biológicos más concretos, con diferentes geometrías y condiciones de contorno.

5.1 Diferentes geometrías

Una aplicación interesante del modelo sería estudiar el efecto del transporte activo en tejidos endoteliales. En ellos, los microtúbulos se disponen en una distribución apical-basal, de modo que sobrerregulando el transporte de los motores moleculares que se dirigen hacia un extremo u otro podría acelerarse o frenarse el transporte de cargos entre membranas.

La geometría de las células endoteliales es aproximadamente ortogonal, así que el cambio no debería suponer ninguna limitación técnica importante, la aplicación ya está preparada para soportar diferentes geometrías en la generación de microtúbulos. Otra geometría interesante a estudiar sería la que encontramos en las neuronas. Aquí el modelo sí que requeriría algunas modificaciones, pues para la subdivisión espacial (Apartado 1.3) se inscribe la geometría en un cubo y se subdivide el cubo completo en celdas, de modo que para geometrías donde la amplia mayoría de las celdas creadas en la subdivisión del cubo no estuviesen contenidas dentro de la célula se estaría asignando una cantidad de memoria inasumible de la cual sólo se estaría utilizando una pequeña parte (ver Figura 19). También relacionado con la geometría de la célula estaría la inclusión del núcleo en el modelo de la célula y el estudio de sus efectos, tanto en el transporte (anisotropías en las zonas con más orgánulos, por ejemplo) como a la hora de generar la red de microtúbulos, que no podría atravesarlo. Esto ya está incluido en la aplicación (Figura 20), aunque no ha sido estudiado.

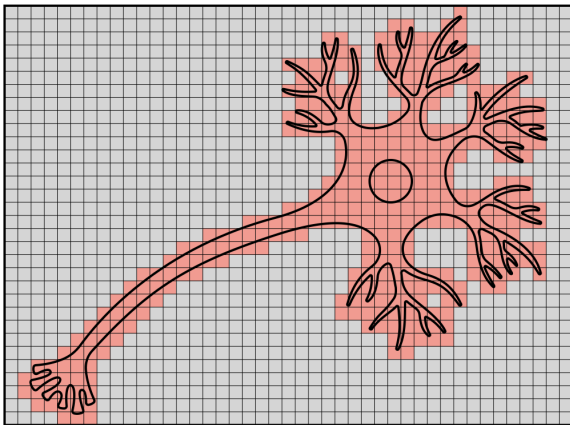


Figura 19. Representación del problema de subdivisión del espacio para una neurona. Las celdas en gris tienen un espacio asignado en memoria, pero no se utilizan nunca. Las celdas en rojo tienen un espacio asignado en memoria y sí se utilizan, pero son comparativamente pocas. Al hacer la subdivisión en 3D y con celdas lo suficientemente pequeñas, la proporción de celdas asignadas y nunca utilizadas se volvería excesivo.

5.2 Microtúbulos flexibles

Los datos de este trabajo han sido recopilados utilizando una estructura de microtúbulos que dejaba de crecer al llegar a la membrana de la célula. Una alternativa sería ir doblándolos progresivamente al acercarse a la membrana celular (o al núcleo), por efecto de la fuerza que estas ejercen sobre los microtúbulos. Una versión preliminar ya está implementada en la aplicación, como se puede apreciar en la Figura 20.

Para modelar la curvatura de los microtúbulos introducida por las membranas, cuando un microtúbulo se encuentra cerca de una membrana (por ejemplo, a menos de 1000 nm) y se calcula que la intersección de no variar la dirección de crecimiento del microtúbulo, se introduce un pequeño *bias* en la dirección en la que se generará el siguiente segmento de microtúbulo para que tenga una componente en dirección opuesta a la membrana. Para calcular esta posible intersección futura y decidir si aplicar o no el *bias* sobre la dirección de crecimiento del microtúbulo se aplica un algoritmo similar al *ray-tracing*.

Este cálculo de la intersección es necesario para crear un modelo visualmente correcto, pues de aplicarlo siempre que un microtúbulo se encuentre cerca de la membrana, se observa que se le induce una curvatura excesiva, pues incluso cuando el microtúbulo ya se está alejando de la membrana en la dirección actual de crecimiento, se sigue aplicando el *bias* si la distancia a la membrana es aún pequeña.

5.3 Combinación de motores moleculares

En una célula hay varios tipos de motores moleculares funcionando al mismo tiempo. Así, habría que estudiar el efecto del trabajo combinado de dineínas y kinesinas y cómo afecta a la distribución de orgánulos y a los tiempos de llegada. En el avance hacia la simulación completa y realista de todo el transporte intracelular es imprescindible emplear varios motores moleculares, pues la regulación al alza o a la baja de la producción de los mismos podría ser uno de los mecanismos que la célula emplease para gestionar la distribución de

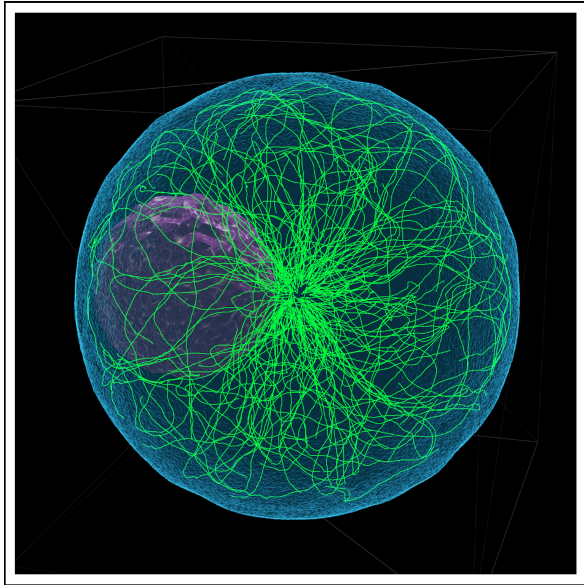


Figura 20. Podría estudiarse el efecto del núcleo en la generación de microtúbulos y su consecuente efecto en el transporte, así como el de microtúbulos que se doblen por la fuerza ejercida por las membranas.

sus orgánulos.

5.4 Modificación del modelo de campo medio

Comprobado el efecto de la suposición de homogeneidad del modelo de campo medio frente a la simulación para microtúbulos perfectamente colineales, sería interesante generar microtúbulos con longitudes de persistencia más bajas y modificar el exponente que determina la densidad de microtúbulos en el modelo de campo medio al computado numéricamente para las estructuras generadas en la simulación, estudiando cómo afecta a las distribuciones de orgánulos y tiempos de llegada y si mejora el ajuste respecto a lo obtenido sin la modificación.

1. Tecnologías utilizadas



Figura 21. De izquierda a derecha y de arriba a abajo: Swift, SceneKit, Metal, Python, NumPy y Matplotlib.

En el desarrollo de la simulación numérica se ha utilizado la librería *SceneKit* [20] de Apple para el renderizado en 3D de alto nivel (visualización de la simulación) y la librería *Metal* [21] de bajo nivel para los *shaders* de computación en GPU (cómputo de la dinámica de las partículas y dinámica de microtúbulos). El resto del código es una aplicación escrita en *Swift* que contiene el código de visualización de la simulación y las llamadas a *Metal* para cálculo en la GPU.

Los resultados de las simulaciones se pueden exportar en un .txt para su procesamiento en Python con las librerías NumPy (cálculo vectorial) y Matplotlib (gráficos).

La elección de Metal en lugar de cualquier otra APIs se debe principalmente a que en la fecha de inicio del trabajo se trataba de la única API de cálculo de bajo nivel en la GPU compatible y activamente mantenida en macOS, dado que el soporte para CUDA había concluido, Vulkan no era compatible y OpenCL llevaba tiempo siendo desaconsejado y dejará de estar soportado en futuras actualizaciones.

2. Diseño de la aplicación

La aplicación para macOS/iPadOS consta de una herramienta de visualización principal donde se puede observar en tiempo real el funcionamiento simulado de la célula: la estructura de microtúbulos, la membrana celular y los orgánulos difundiéndose o siendo transportados por los microtúbulos.

Además, hay una columna vertical con los parámetros modificables de la simulación o una lista de los gráficos que se están obteniendo (también en tiempo real).

En la herramienta de visualización hay una serie de botones que permiten interactuar con la simulación: forzar a que funcione más lentamente (para poder ver mejor la dinámica), desactivar la visualización (para no consumir recursos de la GPU innecesariamente), pausarla o descargar todos los datos de los gráficos en varios .txt, junto a un fichero adicional donde se listan los parámetros utilizados.

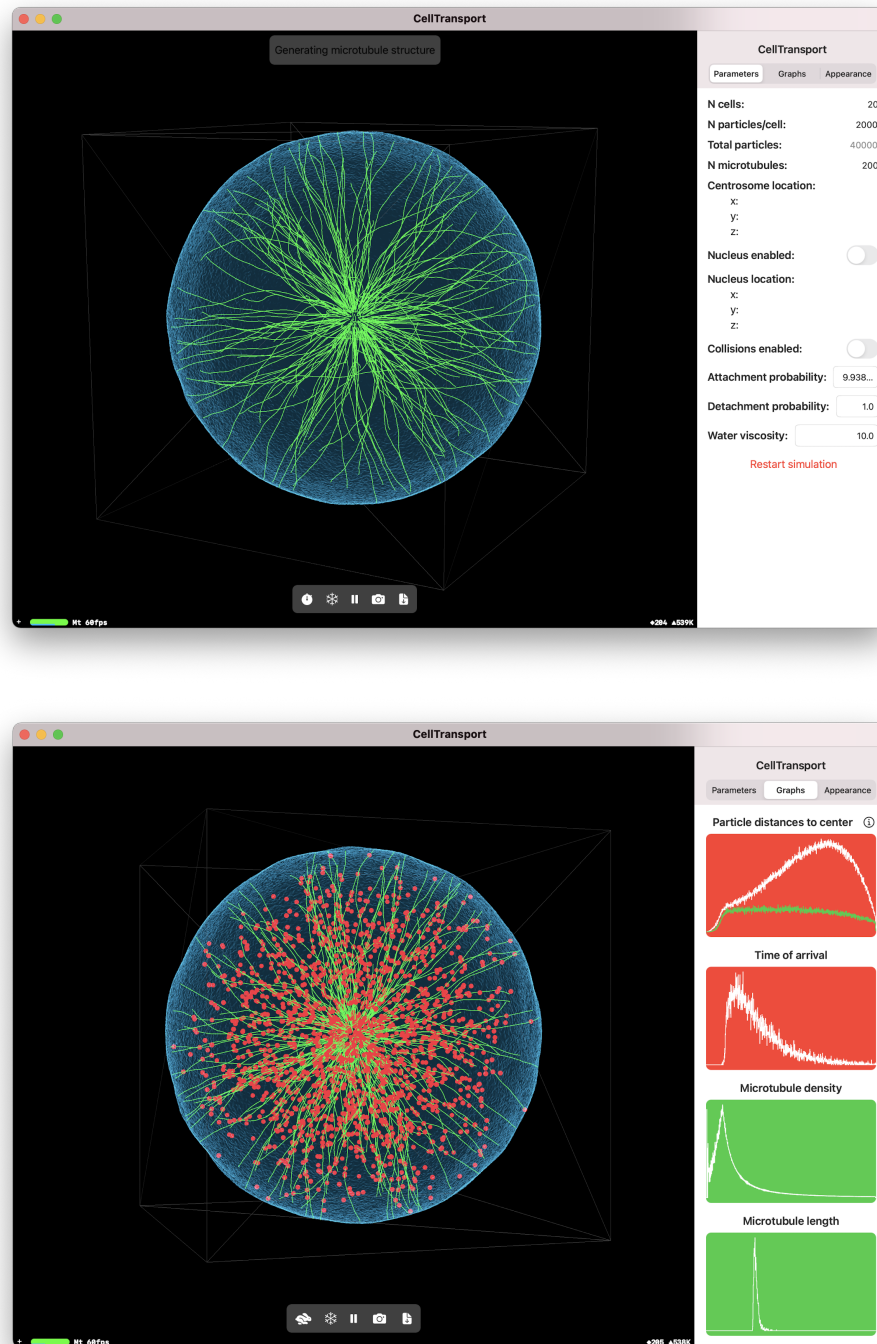


Figura 22. Capturas de pantalla del programa desarrollado y utilizado para la simulación.

3. Arquitectura de la aplicación

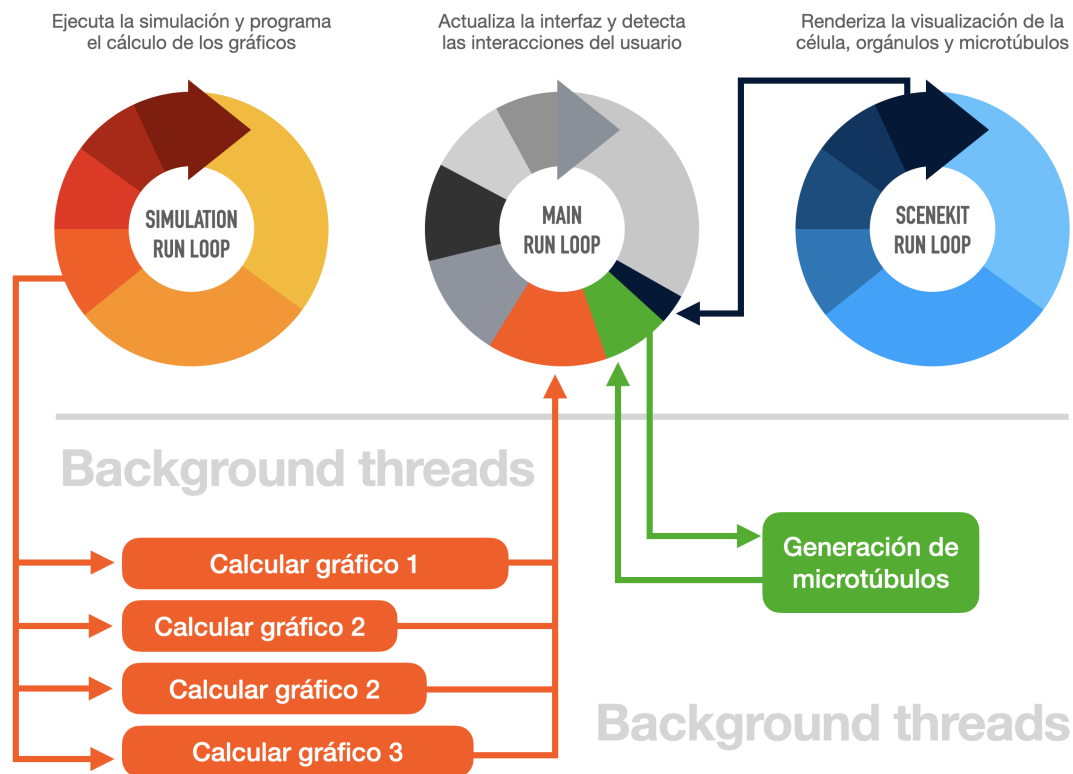


Figura 23. Diseño de la arquitectura de la aplicación. Hay tres *run-loops*, desacoplados entre sí, que funcionan en conjunto para coordinar la interfaz gráfica con la visualización y el cálculo en GPUs.

La arquitectura principal de la aplicación consta de tres *run-loops* (bucles infinitos que procesan eventos) distintos. Por un lado tenemos el loop principal, presente en todas las aplicaciones, que gestiona los cambios en la interfaz y las interacciones del usuario. Este bucle viene dado por el sistema y no tenemos que manipularlo salvo al modificar la interfaz. Para ello, cuando necesitemos un cambio en la interfaz añadimos el código a ejecutar a la *queue* o cola de trabajo del bucle principal y el cambio a realizar será ejecutado en la siguiente iteración del bucle.

Por otro lado tenemos el bucle de *SceneKit*, la librería de representación de gráficos 3D, que también tiene su propio bucle interno. De nuevo, apenas tenemos que modificarlo, y las imágenes renderizadas por el bucle de SceneKit se pasan automáticamente al bucle principal para actualizar la visualización en tiempo real. Este bucle se ejecuta (si es posible) con cada refresco de la pantalla (típicamente 60 o 120Hz).

Y por último, tenemos el bucle de la simulación. Este lo hemos diseñado nosotros, y se ejecuta tan rápido como puede, sin estar limitado por el refresco de la pantalla. En este bucle enviamos los comandos a la GPU y esperamos los resultados (de ahí que sea especialmente importante que no se vea afectado por la velocidad de refresco de la pantalla). Además, una vez calculados los resultados de la GPU, se comprueba, para cada uno de los gráficos de la aplicación (distribución de posiciones, tiempos de llegada, densidad y distribución de longitudes) si ya se está calculando el gráfico o no. De no ser así, se programa en un hilo de fondo el cálculo del gráfico a partir de los resultados de la GPU y se continúa con el bucle de la simulación. Así, como los gráficos se están actualizando en segundo plano, la simulación puede

ejecutarse lo más rápido posible sin tener que esperar a que se dibujen los gráficos cada vez que se obtiene un resultado de la GPU.

Y es este el razonamiento detrás del diseño de los tres bucles: la interfaz, la visualización y el cálculo numérico son procesos independientes, de modo que todos ellos funcionan a la máxima velocidad posible. Así, la interfaz no deja de responder nunca y la visualización se ve fluida incluso cuando el número de partículas simuladas es muy grande y sólo se calculan unos pocos pasos de simulación por segundo. Y también al revés: ni la interfaz ni la visualización entorpecen el cálculo numérico, que va lo más rápido posible en su propio hilo.

Nótese que el bucle o hilo principal también puede programar que una tarea se ejecute en segundo plano sin bloquear la interfaz. Así, la generación de microtúbulos al iniciar la aplicación es programada por el hilo principal, pero se ejecuta en un hilo aparte (de modo que se libera el bucle principal para continuar actualizando la interfaz si fuera necesario) y cuando se ha generado la estructura de microtúbulos se informa al bucle principal de nuevo para que continúe con la ejecución de la aplicación.

4. Algoritmos

D.1 Representación de la membrana celular

La membrana celular se representa visualmente como una esfera del mismo radio que el tomado por la célula en la simulación (normalmente 14000 nm), de un material transparente con reflexión de Fresnel en los bordes.

Para darle un aspecto más natural se crea un ruido de Perlin en 3D de gran escala que se usa como mapa de altura para todos los puntos sobre la membrana. Posteriormente se añade también ruido uniforme para proporcionarle la textura (el ruido Perlin no se podía utilizar en todos los vértices por un bug en el generador de GameplayKit utilizado).

D.2 Representación de microtúbulos

Cada paso temporal los orgánulos tienen una cierta probabilidad de adherirse a los microtúbulos cercanos. A fin de no tener que calcular en cada paso temporal millones de distancias, el espacio está subdividido en celdas de 240 nm de lado. En cada una de estas celdas conocemos qué microtúbulos hay y cuántos son, permitiendo ejecutar la mecánica de adhesión a microtúbulos trabajando sobre estos valores precomputados.

Cada celda tiene un ID asignado y un número de microtúbulos presentes en ella. Cada paso temporal, cuando un orgánulo se mueve, calcula el ID de la nueva celda en la que esté. Con este valor, indexa el array `CellIDToMTIndex`. Si no hay ningún microtúbulo, obtendrá un índice -1 (no válido) que indicará que no puede adherirse a ningún microtúbulo. De haber microtúbulos, obtendrá un nuevo índice y el número de puntos de unión a microtúbulos en la celda.

Si entonces el orgánulo se adhiere a un microtúbulo (después del cálculo de la probabilidad de hacerlo, dependiente del número de microtúbulos) elegirá un número entero al azar entre 0 y el número de microtúbulos de esa celda. Así, indexará el array `CellIDToMTPoint` con la suma del índice obtenido más el número entero entre 0 y el número de microtúbulos de la celda (hacemos esto para que elija un

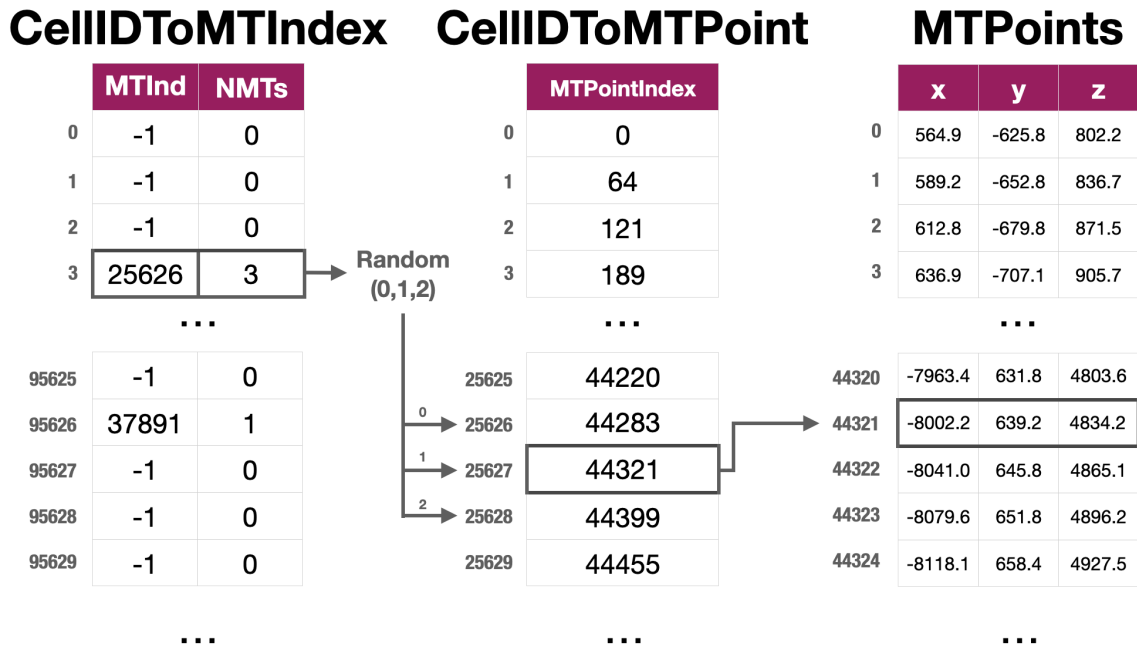


Figura 24. Estructura de los arrays que guardan la posición y el número de microtúbulos para cada celda.

microtúbulo al azar dentro de la celda).

Al indexar CellIDToMTPoint obtendremos un último índice. Con este índice indexamos MTPoints, la lista de todos los puntos de todos los microtúbulos, y conoceremos por tanto la posición del punto de microtúbulo al que adherirnos en el espacio 3D. Para avanzar al siguiente punto del microtúbulo (en un futuro paso temporal) simplemente aumentamos en 1 el índice con el que estamos indexando MTPoints, sin tener que manipular ninguna de las otras matrices. Al final de cada microtúbulo en MTPoints hay un punto en unas coordenadas imposibles que nos indica que hemos llegado al final del mismo.

Podemos ver un resumen de esto mismo en la Figura 24.

5. Generación de números aleatorios

Para la generación de números aleatorios se utiliza una modificación del algoritmo *Mersenne Twister* (*MT-19937*) de 32 bits, de conocido uso en librerías de simulación numérica de todo tipo.

Como estamos trabajando en paralelo, requerimos un generador de números aleatorios independiente para cada hilo. Dado el gran número de hilos con los que trabajamos en la GPU, el tamaño del estado interno del *Mersenne Twister* tradicional (2.5KiB) sería prohibitivo, por lo que pasamos a utilizar *TinyMT*[22], con un estado interno de solo 128 bits, propuesto originalmente en [23], que genera números aleatorios de calidad comparable a *MT-19937* (salvo por el periodo, que pasa a ser de $2^{127} - 1$, antes $2^{19937} - 1$).

Para la paralelización, inicializamos tantos generadores independientes como hilos existan en la *pipeline* de ejecución de la GPU, cada uno con su propio estado interno independiente. La semilla inicial se genera secuencialmente en la CPU para asegurar que los generadores no están correlacionados entre sí, siguiendo el método propuesto en [24]. El estado interno del generador se almacena en un *MTLBuffer* para no tener

que inicializar el generador en cada iteración.

Referencias

1. Faisal, A. A., White, J. A. y Laughlin, S. B. Ion-Channel Noise Places Limits on the Miniaturization of the Brain's Wiring. *Current Biology* **15**, 1143 -1149. ISSN: 0960-9822. <http://www.sciencedirect.com/science/article/pii/S0960982205005713> (2005).
2. Sodeik, B. Mechanisms of viral transport in the cytoplasm. *Trends in microbiology* **8**, 465-72 (nov. de 2000).
3. Höög, J. y col. Organization of Interphase Microtubules in Fission Yeast Analyzed by Electron Tomography. *Developmental cell* **12**, 349-61 (abr. de 2007).
4. Schulze, E y Kirschner, M. Microtubule dynamics in interphase cells. *Journal of Cell Biology* **102**, 1020-1031. ISSN: 0021-9525. eprint: <https://rupress.org/jcb/article-pdf/102/3/1020/1053914/1020.pdf>. <https://doi.org/10.1083/jcb.102.3.1020> (mar. de 1986).
5. Schnitzer, M., Visscher, K. y Block, S. Force production by single kinesin motors. *Nature cell biology* **2**, 718-23 (nov. de 2000).
6. Appert-Rolland, C., Ebbinghaus, M. y Santen, L. Intracellular transport driven by cytoskeletal motors: General mechanisms and defects. *Physics Reports* **593**. Intracellular transport driven by cytoskeletal motors: General mechanisms and defects, 1 -59. ISSN: 0370-1573. <http://www.sciencedirect.com/science/article/pii/S037015731500335X> (2015).
7. Gilbert, T., Le Bivic, A., Quaroni, A y Rodriguez-Boulán, E. Microtubular organization and its involvement in the biogenetic pathways of plasma membrane proteins in Caco-2 intestinal epithelial cells. *The Journal of cell biology* **113**, 275-88 (mayo de 1991).
8. Wolf, K. W. y Spanel-Borowski, K. Acetylation of α -tubulin in different bovine cell types: implications for microtubule dynamics in interphase and mitosis. *Cell Biology International* **19**, 43 -52. ISSN: 1065-6995. <http://www.sciencedirect.com/science/article/pii/S1065699585710062> (1995).
9. Li, J. y col. Estimating Microtubule Distributions from 2D Immunofluorescence Microscopy Images Reveals Differences among Human Cultured Cell Lines. *PloS one* **7**, e50292 (nov. de 2012).
10. Landau, L. y Lifshitz, E. *Statistical Physics: Volume 5* v. **5**. ISBN: 9780080570464. <https://books.google.es/books?id=VzgJN-XPTRsC> (Elsevier Science, 2013).
11. Hsu, H.-P. Monte Carlo Simulations of Lattice Models for Single Polymer Systems. *The Journal of chemical physics* **141**, 164903 (oct. de 2014).
12. Pallavicini, C. y col. Lateral Motion and Bending of Microtubules Studied with a New Single-Filament Tracking Routine in Living Cells. *Biophysical journal* **106**, 2625-2635 (jun. de 2014).
13. Wisanpitayakorn, P., Mickolajczyk, K. J., Hancock, W. O., Vidali, L. y Tüzel, E. Measurement of the Persistence Length of Cytoskeletal Filaments using Curvature Distributions. *bioRxiv*. eprint: <https://www.biorxiv.org/content/early/2018/01/23/252551.full.pdf>. <https://www.biorxiv.org/content/early/2018/01/23/252551> (2018).
14. Boal, D. *Mechanics of the Cell* 2.^a ed., 89 (Cambridge University Press, 2012).

15. Bressloff, P. y Newby, J. Stochastic models of intracellular transport. *Reviews of Modern Physics* **85** (ene. de 2013).
16. Feng, Q., Mickolajczyk, K., Chen, G. y Hancock, W. Motor Reattachment Kinetics Play a Dominant Role in Multimotor-Driven Cargo Transport. English (US). *Biophysical Journal* **114**, 400-409. ISSN: 0006-3495 (ene. de 2018).
17. Press, W. H., Teukolsky, S. A., Vetterling, W. T. y Flannery, B. P. *Numerical Recipes in C: The Art of Scientific Computing. Second Edition* (1992).
18. Luby-Phelps, K. en *Microcompartmentation and Phase Separation in Cytoplasm* (eds. Walter, H., Brooks, D. E. y Srere, P. A.) 189 -221 (Academic Press, 1999). <http://www.sciencedirect.com/science/article/pii/S0074769608605276>.
19. Spreng, B y col. Microtubule number and length determine cellular shape and function in Plasmodium. *The EMBO Journal* **38**, e100984 (2019-08-01 00:00:00.0).
20. Apple. *SceneKit* <https://developer.apple.com/documentation/scenekit/> (2020).
21. Apple. *Metal* <https://developer.apple.com/documentation/metal> (2020).
22. Saito, M. y Matsumoto, M. *Tiny Mersenne Twister (TinyMT): A small-sized variant of Mersenne Twister* <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/TINYMT/index.html> (2020).
23. Saito, M. y Matsumoto, M. A high quality pseudo random number generator with small internal state. 研究報告ハイパフォーマンスコМПユテイング (HPC) **2011**, 1-6. <https://ci.nii.ac.jp/naid/110008620834/en/> (2011).
24. Matsumoto, M. y Nishimura, T. *Dynamic Creation of Pseudorandom Number Generators* en *Monte-Carlo and Quasi-Monte Carlo Methods 1998* (eds. Niederreiter, H. y Spanier, J.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2000), 56-69. ISBN: 978-3-642-59657-5.